UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/850,382	05/07/2001	Gerard J. Holzmann	Holzmann 18	2715

47394 7590 01/21/2005

HITT GAINES, PC
LUCENT TECHNOLOGIES INC.
PO BOX 832570
RICHARDSON, TX 75083

EXAMINER	
YIGDALL, MICHAEL J	
ART UNIT	PAPER NUMBER
2122	

DATE MAILED: 01/21/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	**Application No.** 09/850,382	**Applicant(s)** HOLZMANN, GERARD J.	
	Examin r Michael J. Yigdall	**Art Unit** 2122	

-- The MAILING DATE of this communication app ars on the cover she t with th correspond nce address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *17 September 2004*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-16* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-16* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. Applicant's request for reconsideration filed on September 17, 2004 has been fully considered. Claims 1-16 are pending.

Response to Arguments

2. Applicant's arguments have been fully considered but they are not persuasive.

3. Applicant contends that Edwards (and likewise, Holzmann) does not teach or suggest extracting a verification model from source code comprising generating the verification model in a target language including populating a control flow of procedures in the source code with strings of the target language (Applicant's remarks, page 4).

However, Edwards discloses generating a model, in the form of a hardware representation, based on a flow graph (see, for example, column 6, lines 26-42). The flow graph includes control flow paths from the source code (see, for example, column 5, line 62 to column 6, line 2), and is populated with nodes or "strings" of the hardware representation (see, for example, column 6, lines 15-25). The flow graph and hardware representation are "extracted" from source code specifications (see, for example, column 3, lines 36-39). Importantly, Edwards expressly discloses that the hardware representation is a target language (see, for example, column 5, lines 39-44).

4. In response to Applicant's argument that there is no suggestion to combine the references (Applicant's remarks, page 5), the examiner recognizes that obviousness can only be established by combining or modifying the teachings of the prior art to produce the claimed invention where there is some teaching, suggestion, or motivation to do so found either in the references

themselves or in the knowledge generally available to one of ordinary skill in the art. See *In re Fine*, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988) and *In re Jones*, 958 F.2d 347, 21 USPQ2d 1941 (Fed. Cir. 1992).

In this case, Edwards discloses optimizing the model with logic reduction (see, for example, column 8, lines 42-54). Holzmann similarly discloses optimizing a model with state space reduction (see, for example, column 2, lines 15-23), so as to verify the correctness of complex system designs based on properties to be verified (see, for example, column 1, lines 16-35). One of ordinary skill in the art would have been motivated to verify the correctness of a model generated by the method of Edwards. For example, the method of Edwards may be used to design a system for an integrated circuit (see, for example, column 1, lines 28-31). Holzmann discloses that such integrated circuits must operate correctly and are expected to work perfectly (see, for example, column 1, lines 16-24). Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to supplement Edwards with the verification features taught by Holzmann, so as to verify the correctness of the model.

Double Patenting

5. The provisional rejection of claims 1-16 under the judicially created doctrine of obviousness-type double patenting as being unpatentable over claims 1-23 of copending Application No. 09/809,499 is withdrawn in view of the timely filed terminal disclaimer in compliance with 37 CFR 1.321(c).

6. The rejection of claims 1-16 under the judicially created doctrine of obviousness-type double patenting as being unpatentable over claims 1-23 of U.S. Pat. No. 6,353,896 is withdrawn in view of Applicant's remarks (pages 2-3, section II).

Claim Rejections - 35 USC § 103

7. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

8. Claims 1-9 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Pat. No. 6,625,797 to Edwards et al. (art of record, "Edwards") in view of U.S. Pat. No. 5,615,137 to Holzmann et al. (art of record, "Holzmann").

With respect to claim 1 (original), Edwards discloses a method for extracting a verification model from source code (see, for example, the abstract and FIG. 1) comprising the steps of:

defining a control flow for procedures in the source code (see, for example, column 3, lines 36-39, which shows extracting information from a source code specification and defining control flow paths for functions or procedures in the source code);

generating source strings for selected elements of the source code (see, for example, column 6, lines 4-14, which shows generating sequences of source bytecodes, i.e. source strings);

associating the source strings to an interpretation according to a plurality of prioritized mapping rules (see, for example, column 6, lines 15-25, which shows associating the source bytecodes or strings with nodes of a flow graph based on a specification, and column 3, lines 22-35, which further shows annotating the nodes with mapping rules for the implementation);

applying the associated interpretation to the source strings to translate the source strings to strings of a target language (see, for example, column 5, lines 39-44, which shows applying the flow graph representation to translate the source code to a target language);

generating the verification model in the target language, the generating step including the step of populating the control flow with the strings of the target language, wherein the verification model conforms to the control flow (see, for example, column 6, lines 26-42, which shows generating the implementation or model based on the flow graph, and column 5, line 62 to column 6, line 2, which shows populating the graph with the nodes and control flow paths).

Although Edwards discloses optimizing the model with techniques such as logic reduction (see, for example, column 8, lines 42-54), Edwards does not expressly disclose the step of:

optimizing the verification model according to a property to be verified.

However, Holzmann discloses a system for optimizing a model with state space reduction according to properties to be verified (see, for example, the abstract and column 2, lines 15-23), so as to verify the correctness of complex systems such as integrated circuits (see, for example, column 1, lines 16-35).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to supplement the optimizations disclosed by Edwards with those taught by

Holzmann, for the purpose of verifying the correctness of the model according to properties to be verified.

With respect to claim 2 (original), Edwards further discloses the limitation wherein the plurality of mapping rules comprises at least one explicit mapping (see, for example, column 9, lines 21-25 and column 18, lines 11-16, which show examples of explicit mappings).

With respect to claim 3 (original), Edwards further discloses the limitation wherein the plurality of mapping rules comprises at least one data restriction (see, for example, column 10, lines 46-57, which shows a data restriction based on data type precision).

With respect to claim 4 (original), Edwards further discloses the limitation wherein the plurality of mapping rules comprises at least one default type rule (see, for example, column 10, lines 35-46, which shows defaulting to the maximum data type precision).

With respect to claim 5 (original), Edwards further discloses the limitation wherein the plurality of mapping rules comprises at least one explicit mapping, at least one data restriction and at least one default type rule (see, for example, column 9, lines 21-25 and column 18, lines 11-16, which show examples of explicit mappings; also see, for example, column 10, lines 46-57, which shows a data restriction based on data type precision; also see, for example, column 10, lines 35-46, which shows defaulting to the maximum data type precision).

With respect to claim 6 (original), Edwards further discloses the limitation wherein associating the source strings to an interpretation according to a plurality of prioritized mapping rules comprises the further steps of, for each source string:

(a) searching a lookup table for an explicit mapping that matches the source string (see, for example, column 9, lines 21-25 and column 18, lines 11-16, which show examples of explicit mappings that match portions of the source code; note that a lookup table or some other analogous data structure is inherently searched for such mappings);

(b) if a matching explicit mapping is found in step (a), associating the source string to the interpretation corresponding to the explicit mapping (see, for example, column 6, lines 15-25, which shows associating the source bytecodes or strings with nodes representing the target implementation, i.e. corresponding to the explicit mappings);

(c) if no matching explicit mapping is found in step (a), determining if a data restriction applies to the source string (see, for example, column 10, lines 26-32, which shows determining a data restriction, e.g. precision, when there is no explicit definition or mapping);

(d) if a single applicable data restriction is determined in step (c), associating the source string to the interpretation corresponding to the single applicable data restriction (see, for example, column 10, lines 46-57, which shows applying the data restriction to the nodes from the source code);

(e) if a plurality of applicable data restrictions are determined in step (c), selecting one of the applicable data restrictions and associating the source string to the interpretation corresponding to the selected data restriction (see, for example, column 10, line 58 to column 11, line 7, which shows selecting and applying one data restriction from a plurality of data restrictions);

(f) if no applicable data restriction is found in step (c), associating the source string to the interpretation according to a default type rule (see, for example, column 10, lines 35-46, which shows applying a default data type rule, e.g. precision, when a data restriction has not been found).

With respect to claim 7 (original), Edwards further discloses the limitation wherein the lookup table contains source string patterns representing a plurality of entries in the lookup table, wherein searching the lookup table includes searching for the source string patterns (see, for example, column 10, lines 9-14, which shows detecting keywords in the source code, i.e. source string patterns, to identify a mapping to the target implementation).

With respect to claim 8 (original), Edwards further discloses the limitation wherein the application of the mapping rules causes the translating of the source strings to respective equivalent statements in the target language when the selected source code elements are fully relevant to a property to be tested and the translating of the source strings to nul statements in the target language when the selected source code elements are irrelevant to the property to be tested (see, for example, column 8, line 56 to column 9, line 10, which shows reducing the flow graph so that only those source code elements relevant to the target implementation will be translated; note that because null statements are analogous to no-op statements, it would have been obvious to one of ordinary skill in the art at the time the invention was made to translate the irrelevant source code elements to null or no-op statements rather than not including them in the model).

With respect to claim 9 (original), Edwards further discloses the limitation wherein the source code is selected from the group comprising C, C++, and Java (see, for example, column 6, lines 4-14, which shows that the source code may be written in the C, C++ or Java languages).

9. Claims 10-12, 15 and 16 are rejected under 35 U.S.C. 103(a) as being unpatentable over Holzmann in view of Edwards.

With respect to claim 10 (original), Holzmann discloses, in a computer-based model checker, a method for automatically verifying a property of a system using the system source code, the model checker operable to check a verification model for the property (see, for example, column 3, lines 38-64, which shows a model checker for verifying a property of a system).

Although Holzmann discloses modeling a system for verification (see, for example, the abstract) and inputting both the system description, i.e. the verification model, and a representation of the property to the model checker (see, for example, column 3, lines 38-64), Holzmann does not expressly disclose additional steps recited in the claim.

However, Edwards discloses a method for extracting a model of a system (see, for example, the abstract and FIG. 1) comprising the steps of:

inputting the source code, a conversion table, a representation of the property and an optional preferences file to the apparatus, the conversion table including strings corresponding to strings of the source code and interpretations mapped to the strings, the preferences file including interpretations for overriding default rule interpretations (see, for example, column 6, lines 4-14, which shows inputting the source code, and column 3, lines 22-35, which shows inputting annotations or preferences for overriding default implementations; also see, for example, column 9, lines 21-25 and column 18, lines 11-16, which show examples of conversions corresponding to portions of the source code to mapped to implementations; note that a conversion table or some other analogous data structure is inherently provided for such conversions);

programming the model checker with default rule interpretations, wherein the default rule interpretations when applied by the model checker translate source code strings to a language of the model checker (see, for example, column 6, lines 15-25, which shows generating a flow graph representation based on a specification, i.e. default rules, and column 5, lines 39-44, which shows applying the representation to translate the source code to a target language);

defining a control flow for each procedure in the source code (see, for example, column 3, lines 36-39, which shows extracting information from a source code specification and defining control flow paths for functions or procedures in the source code);

selecting source code strings for translation from the source code to the language of the model checker (see, for example, column 6, lines 4-14, which shows selecting source code sequences or strings for translation);

for each selected string, according to a predetermined priority:

searching the conversion table for entries corresponding to the selected string (see, for example, column 9, lines 21-25 and column 18, lines 11-16, which show examples of conversions corresponding to portions of the source code; note that a conversion table or some other analogous data structure is inherently searched for such entries);

translating the selected string according to the interpretation mapped to the selected string (see, for example, column 6, lines 15-25, which shows translating the source code to a corresponding flow graph having nodes mapped to the source sequence or string);

applying the default rule interpretation corresponding to the selected string (see, for example, column 6, lines 15-25, which shows applying a specification, i.e. default rules); and

overriding the default rule interpretation according to an entry in the preferences file (see, for example, column 6, lines 26-24, which shows applying constraints and preferences to override the default rules);

populating the control flow with the interpretations to provide the verification model (see, for example, column 6, lines 26-42, which shows generating the implementation or model based on the flow graph, and column 5, line 62 to column 6, line 2, which shows populating the graph with the nodes and control flow paths).

Holzmann further discloses the step of:

checking the verification model for the property (see, for example, column 3, lines 38-64, which shows checking the system description, i.e. the verification model, for the property).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to use the model extraction features taught by Edwards in the model checking system of Holzmann, for the purpose of automatically generating a model of a system with which to optimize and verify the design (see, for example, Edwards, column 1, line 62 to column 2, line 2, and Holzmann, column 1, lines 16-35).

With respect to claim 11 (original), Holzmann discloses a computer-based model checker (see, for example, column 3, lines 38-64) comprising:

a processor for executing instructions (note that a processor is inherently used to execute instructions in the model checking system);

storage accessible to the processor for storing the instructions, a lookup table, default rules, source code of a system, a property to be verified and an optional preferences file (note that storage is inherently accessible to the processor in the model checking system).

Although Holzmann discloses modeling a system for verification (see the abstract) and providing a property to be verified (see, for example, column 3, lines 38-64), Holzmann does not expressly disclose additional features recited in the claim.

However, Edwards discloses a system for extracting a model of a system (see the abstract and FIG. 1) comprising program code or instructions (see, for example, column 21, lines 30-34), the instructions causing the processor to:

parse the source code and to define a control flow for procedures in the source code (see, for example, column 4, lines 40-45, which shows parsing the source code, and column 3, lines 36-39, which shows extracting information from a source code specification and defining control flow paths for functions or procedures in the source code);

generate source strings for selected source code elements (see,.for example, column 6, lines 4-14, which shows generating source code sequences or strings);

selectively associate the source strings to an interpretation according to a plurality of mapping rules, including mapping rules defined in the lookup table, in the default rules and in the optional preferences file (see, for example, column 6, lines 15-25, which shows associating the source strings with nodes of a flow graph according to a specification, i.e. default rules, and column 3, lines 22-35, which shows annotations or preferences applied to the association; also see, for example, column 9, lines 21-25 and column 18, lines 11-

16, which show examples of mapping rules; note that a lookup table or some other analogous data structure is inherently defined for such mappings);

apply the associated interpretation to the source strings to translate the source strings to strings which can be operated on by the model checker (see, for example, column 5, lines 39-44, which shows applying the flow graph to translate the source code to a target language);

populate the control flow with the strings, the populated control flow being a verification model (see, for example, column 6, lines 26-42, which shows generating the implementation or model based on the flow graph, and column 5, line 62 to column 6, line 2, which shows populating the graph with the nodes and control flow paths).

Holzmann further discloses causing the processor to:

check the verification model for the property (see, for example, column 3, lines 38-64, which shows checking the system description, i.e. the verification model, for the property); and

an output device responsive to the processor for providing a result of the check (see, for example, column 3, lines 38-64, which shows outputting a result of the check; note that an output device is inherently used to provide the output).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to use the model extraction features taught by Edwards in the model checking system of Holzmann, for the purpose of automatically generating a model of a system with which to optimize and verify the design (see, for example, Edwards, column 1, line 62 to column 2, line 2, and Holzmann, column 1, lines 16-35).

With respect to claim 12 (original), Holzmann further discloses the limitation wherein the model checker is a SPIN model checker (see, for example, column 3, lines 50-52, which shows a SPIN model checker).

With respect to claim 15 (original), Holzmann in view of Edwards further discloses the limitation wherein the lookup table includes entries corresponding to branch conditions (see, for example, Edwards, column 12, line 61 to column 13, line 12, which shows conditional branches and thread branches applied to the flow graph).

With respect to claim 16 (original), Holzmann in view of Edwards further discloses the limitation wherein the entries corresponding to branch conditions include entries for introducing a nondeterministic choice to the verification model (see, for example, Edwards, column 12, line 61 to column 13, line 12, which shows that the thread branches provide multiple control flow outputs without input data, introducing a nondeterministic choice).

10. Claims 13 and 14 are rejected under 35 U.S.C. 103(a) as being unpatentable over Holzmann in view of Edwards as applied to claim 11 above, and further in view of U.S. Pat. No. 6,389,385 to King (art of record, "King").

With respect to claim 13 (original), Holzmann in view of Edwards does not expressly disclose the limitation wherein the interpretations comprise print, hide, comment and keep, wherein print embeds the source string into a print action of the model checker, hide excludes the source string from representation in the verification model, comment includes the source string in the verification model as a comment, and keep preserves the source string in the verification model.

However, King discloses a system for translating source code using a mapping table (see, for example, the abstract and block 74 in FIG. 2), wherein source characters or strings are translated and printed to the model, replaced with markers and excluded from the model, or included and preserved as comments (see, for example, FIG. 2 and column 3, lines 10-31).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to supplement the model checking and extraction system of Holzmann and Edwards with print, hide, comment and keep interpretations having the functionality taught by King, for the purpose of enabling safe and reversible translations (see, for example, King, column 1, line 66 to column 2, line 4) of source code to verification models.

With respect to claim 14 (original), Holzmann in view of Edwards in view of King further discloses the limitation wherein the keep preserves the source string in the verification model subject to global substitute rules (see, for example, King, FIG. 2 and column 3, lines 10-31, which shows checking the mapping table, i.e. for global substitute rules, to determine whether to preserve the source characters or strings).

Conclusion

11. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO

MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

12. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Michael J. Yigdall whose telephone number is (571) 272-3707. The examiner can normally be reached on Monday through Friday from 7:30am to 4:00pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam can be reached on (571) 272-3695. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

MY

Michael J. Yigdall
Examiner
Art Unit 2122

mjy

TUAN DAM
SUPERVISORY PATENT EXAMINER